# Many Models in R

## Transcript video 3

Full resource: <https://www.ncrm.ac.uk/resources/online/all/?id=20840>

Okay, so now we've done the first two steps. We've created a specification grid in the first steps.

Let's have a quick look at mod specs. In the second step we've taken a specification ID and we've created a function that returns the results of a regression model just based on that ID alone. So for instance we can do getln1 which gives us the result of the first model.

Now we can go on to the third step which is to run the regression model for each of the specifications within the model specification grid. Obviously we don't want to just run out getln96 times as that would take far too long. Instead we can use the map function which is from the per package which is within the tidyverse to do this repetitive step. Now map has a unique syntax so I'm going to quickly demonstrate how it works using a toy example.

Map is an iterator function. It takes an object and applies a function to each element of that object returning the results of the list. Let's have a quick look. So here we're going to input the vector 2 to 5 so that's 2, 3, 4 and 5 and we're just going to add 16 to each of those numbers. So we get back a list of length 4, 18, 19, 20 and 21, 2 plus 16, 3 plus 16, 4 plus 16 and 5 plus 16.

So the way that I've told map to do this function, this plus 16 function, is using something called lambda notation. We have this tilde and this dot X placeholder which is just a placeholder for the current value of the object being looped over. So in the first case dot X is 2, in the second case is 3, then 4, then 5 and that's why we have the results 18, 19, 20 and 21 stored as a list.

We can use map to loop over the set of spec IDs that we have in our model specification grid and run getlm for each one. Modspecs$specID is just a vector going from 1 to 96 so we can loop over that in the same way that we did just adding 16 in the previous step.

So here we'll just run getlm on each of these numbers. Now that's going to take a little time to run but you'll see that it's returned a list of length 96 because we've got 96 different models and this last element is the result for the very last specification.

This 95th one is the result for the penultimate specification and so on. Now getting back the data in this format isn't very helpful, one because it's in a list and two because it doesn't tell us what model specifications these results correspond to. We can't easily see what the specification was for this particular regression that was run so what does this 96th result actually mean? Instead we can use a very neat feature of data frames within R which is something called a list column.

So let's just look at modspecs again. At the minute modspecs is just a standard data frame where we have five different columns and each of these columns is a vector.

In R a vector is a set of values that are of the same type so for instance specID is a list of integers, covars is a list of character vectors or strings, that's also true of cogvar, fup is an integer and then cohort is a factor variable.

So what we can do is have a list as a column within a data frame too and as I say map returns a list so we can put the result of a map function right into this modspecs data frame by using map within a mutate command. Mutate is used to create new columns. So here we'll just create a new column called res which is going to contain the results of this 96 regression models. So let's just quickly run that now.

Okay it'll take a little while to run again. Right so now you see we have this new column res which is a list of tables. Each of them has one row and five columns like dimensions 1 by 5 and that's because each of them is just a table like this. It's returned by the getlm function. Now this doesn't work very well because we can't see the results directly we can just see we've got this list of tables.

So instead we can use the unnest function to explode out the res column into the wider data frame. So instead of having a list of tables we'll have five new columns containing the information that was in res. So estimate, standard error, p-value, confidence interval, low and high.

So let's just run that again and add an unnest function. Okay there we go. We've still got 96 rows in the original five columns but we also now have the extra five variables estimate, standard error, p-value, conflow and conf high.

You can't see conf high but it's just printed there. We can see that the association age 15 for baby boomers using the nonverbal cognitive ability variable and the basic set of covariates was minus 0.169. Let's just save this as mod pres and let's just show that off again. And there we have it we've got the results of 184 regressions. It was very simple to run.

Thank you.

National Centre for Research Methods (NCRM)  
Social Sciences  
Murray Building (Bldg 58)  
University of Southampton  
Southampton SO17 1BJ  
United Kingdom

**Web** www.ncrm.ac.uk   
**Email** info@ncrm.ac.uk  
**Tel** +44 23 8059 4539  
**Twitter** @NCRMUK